

Simulation and Implementation of BPSK BPTC and MSK Golay Code in DSP Chips

Yihua Chen, Juehsuan Hsiao, PangFu Liu, Kunfeng Lin

Oriental Institute of Technology Institute of Information and Communication Engineering New Taipei city, Taiwan
ff011@mail.oit.edu.tw, ff014@mail.oit.edu.tw, mars7699@gmail.com

Abstract—This study discussed the decoding mechanism of block product turbo code (BPTC) (196,96) and simulated its efficiency, used the Hamming (15,11) and Hamming (13,9) block channel code combinations and block interleaving to construct a BPSK modulation and BPTC (196,96) coding system in the concept of feedback encoding in turbo code. The decoding mechanism was meticulously described to explain the advantage of this turbo feedback structure. This study also analyzed and simulated special combination of MSK modulation with Golay (20, 8) channel code. Golay (20,8) decoding correction capability was comprehensively discussed. The simulation results indicate MSK Golay (20,8) is less robust than BPSK Golay (20,8) for 2.8dB against noise, but BPSK BPTC (196, 96) is 3dB more robust than MSK Golay (20, 8) and it is also a better choice than BPSK Golay (20, 8), if signal E_b/N_0 is stronger than 5dB. Whenever the C code system program is constructed, the system program should be optimized and realized on the DSP platform. Both MSK Golay and BPSK BPTC C program optimization efficiency statistics are analyzed. The implementation can be applied to the wireless communication system of SDR concept, so that different modulation and encoding modes can adjust parameters in firmware directly according to different system specification requirements.

Keywords—component; BPSK; MSK; Golay Code; BPTC; Coding Gain; Syndrome; Perfect Code; Error Pattern; BER; SDR;

I. INTRODUCTION AND BACKGROUND

When a wireless communication system is in a non-Line-of-sight (NLOS) environment, the signal will be refracted, reflected, diffracted and scattered when passing through different features in the course of transmission. For a real-time communication system, how to reduce the error probability of digital signal in propagation is a very important issue. The channel coding is a technology that reduces the signal distortion resulted from multi-path interference and AWGN noise in receiving. Since the channel coding must add excess bits to the original data in transmitting (except Trellis Coded Modulation, TCM), called parity bit, so that we can detect and correct the error bits of signal. The additional parity bits will reduce the energy of received signal symbol, so that the symbol error rate in receiver increases. However, if more error bits can be corrected to compensate for the symbol error rate increased by the decrease in symbol energy, the bit error rate of the whole received signal still can be reduced. If only the same bit error rate is required to be kept, the power efficiency E_b/N_0 of the received digital signal after decoding can be reduced. In other words, the coding gain is increased [1], the precondition is that the bit error rate must be identical.

$$G(\text{dB}) = (E_b/N_0)_{\text{uncoding}}(\text{dB}) - (E_b/N_0)_{\text{coding}}(\text{dB})$$

To attain the coding gain, the signal-to-noise (SNR) of the received signal must reach a certain E_b/N_0 threshold, otherwise the addition of coding parity bits increase the symbol error rate of the received signal. The decoding mechanism cannot compensate for this loss, and there will be no coding gain at all. Therefore, the selection of coding rate is very important; in other words, adopt different encoding modes and different coding rates should be adopted according to different system specification requirements. The coding rate is determined by the number of information bits and parity bits that influences the decoding capability of channel code significantly.

The bit error rate of signal in transmission can actually be improved by adding channel coding in a wireless channel environment. The excess bandwidth is taken out from limited bandwidth for channel coding to make the receiver have lower bit error rate at the same transmitting power. Alternatively, the required communication quality can be attained at lower transmitted power, from which we have coding gain of the channel code. Both benefits can only be achieved when the E_b/N_0 threshold of received signal are reached. The E_b/N_0 threshold effect will be discussed in this study.

In a wireless channel environment, the most frequent problem is the interference and noises from different environments, such as atrocious weathers, obstacles and user moving. Therefore, continuous burst errors are likely to happen, that will cause great difficulties in coding. Therefore, we must use block interleaving to break up the burst errors into discontinuous random errors, so as to exert the capability of channel coding to correct errors.

Generally, adding more redundancy bits can correct more random error bits, but the difficulty in coding and decoding will increase, and the complex decoding algorithm requires a larger memory and complex software and hardware design [2].

The turbo code was introduced by Berrou, Glavieux and Thitimajshima in 1993, published in literatures [3] [4] which indicate that this scheme can reach the bit error probability of 10^{-5} in additive white Gaussian noise channel and E_b/N_0 of 0.7 dB BPSK modulation at 1/2 code rate. This encoding mode combines more than two simple block codes or convolutional codes for larger coding gain. The concept behind turbo decoding is to transmit the soft decision from the output end of the first decoder to the input end of another decoder, and iterate this process for several times, so as to generate relatively reliable decisions and add in interleaving coding.

In 1948, Hamming put forward an important error-correcting code, named Hamming code [5]. It uses parity check matrix to detect and correct errors; however, its detection and

correction capability is limited, it can only detect 2-bit errors or correct 1-bit errors.

This paper discusses the block product turbo code (BPTC) which is classified as one of block turbo code concatenation forms. The Hamming code can detect two-bit error or correct one bit error. The BPTC uses two Hamming codes for "column" coding and "row" coding, it has improved the Hamming code correcting only one error. In addition, the BPTC carries out block interleaving coding for disorganizing the transmission sequence before transmission, so as to avoid burst errors when the signal meets multi-path channel in the channel. This paper will also explain the principle of error bit correcting mechanism of BPTC channel coding.

There are six sections in this paper. The second section introduces the operating principle of the two Hamming codes, Hamming (15, 11) and Hamming (13, 9), used in BPTC code. The third and fourth sections discuss the BPTC and Golay (20, 8) code design principles. In the fifth section, we discuss all the simulated data of BPSK BPTC and MSK Golay. In the last section we infer some conclusions of these results.

II. HAMMING CODE

A. Hamming Encoding

The Hamming code is a sort of binary linear system block code (as shown in Fig. 1). Hamming put forward an important single-error-correcting code, using parity check matrix (H) to detect and correct errors. Hamming code is an important forward error correction (FEC) in theory and practice so far. Furthermore, Hamming code (7, 4) and (16, 11) is very limited perfect code in systematic code. But is not for Hamming (15, 11) and Hamming (13, 9), used in this paper.

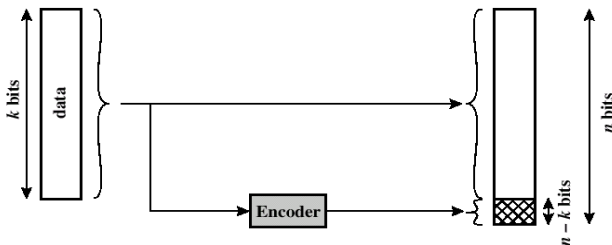


Figure 1. Schematic diagram of Hamming coding process

Hamming code is a simple type of systematic code, described as the following structure.

Block length: $n = 2^m - 1$

Number of data bits: $k = 2^m - m - 1$

Number of check bits: $n - k = m$

Minimum distance: $d_{\min} = 3 \rightarrow$ Correct single bit error

$(n, k) = (2^m - 1, 2^m - 1 - m)$

Generally speaking, we can use the following $k \times n$ array to define the generator matrix (G)

$$G = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_k \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ V_{k1} & V_{k2} & \cdots & V_{kn} \end{bmatrix}$$

Among which, $V_1 \sim V_k$ are linearly independent vectors that can generate all code vectors. The data of the transmitting terminal are usually expressed in column vectors, therefore, the sequence of k message bits, i.e. the message m is expressed as $1 \times k$ matrix. If we use matrix notation to indicate the generation process of code word, i.e. the product of m and G ,

$$U = mG$$

The two Hamming codes used in this paper are Hamming (15, 11) and Hamming (13, 9), as defined above, the (15, 11) code means n is 15 and k is 11, the redundancy bit added in the coding process is 4, and the correction capability is 1. The (13, 9) code means n is 13, k is 9, the redundancy bit added in the coding process is 4, and the correction capability is 1.

The generator matrices are shown in Fig. 2 and Fig. 3.

1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1

Figure 2. Generator matrices Hamming (15, 11)

1	0	0	0	0	0	0	0	0	0	1	1	1	1
0	1	0	0	0	0	0	0	0	0	1	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	0	0	1	0	0	0	0	0	0	1	0	1	0
0	0	0	0	1	0	0	0	0	0	1	0	1	1
0	0	0	0	0	1	0	0	0	0	1	1	0	0
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	0	1	1

Figure 3. Generator matrices Hamming (13, 9)

B. Hamming Decoding

In order to decode the received signals, we need to define a parity check matrix and a syndrome. There is a $(n-k) \times n$ matrix H in each generator matrix G , so that the columns of G are orthogonal to the columns of H , i.e. $GH^T = 0$, the H^T is the transpose matrix of H . In order to meet the orthogonality of system coding, the component of matrix H can be expressed as

$$H = [I_{n-k} \mid P^T]$$

Therefore, the matrix H^T can be expressed as

$$H^T = \begin{bmatrix} I_{n-k} \\ P^T \end{bmatrix}$$

U is the code word derived from matrix G if and only if $UH^T = 0$. Let r be the vector received by the receiving terminal, so the r can be described as

$$r = U + e$$

Among which, e is the error vector resulted from the channel. There are $2^n - 1$ error patterns in nonzero position in the space formed by 2^n value groups. The syndrome is defined as

$$S = rH^T$$

The syndrome is the result of parity check implemented in r, judge whether r is an effective element in the codeword set. Based on development of equation

$$S = (U + e)H^T = UH^T + eH^T$$

However, for all elements in codeword set $UH^T = 0$, therefore

$$S = eH^T$$

Since the correction capability of Hamming(15,11) and Hamming(13,9) is 1, meaning the error pattern is one selected from n, so the syndrome tables can be set up, the syndrome tables are shown in Fig. 4.

syndrome	error pattern	syndrome	error pattern
1001	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1111	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1101	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	1110	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1111	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	0111	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
1110	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	1010	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0111	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	0101	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1010	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	1011	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0101	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	1100	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1011	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	0110	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
1100	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	1000	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0110	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	0100	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0011	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0	0010	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
1000	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	0001	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0100	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	0000	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0010	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	0000	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0001	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	1001	0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
0000	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1101	0 0 0 0 0 0 0 0 0 1 0 0 0 1

(a.) Hamming (15, 11)

(b.) Hamming (13, 9)

Figure 4. Hamming code syndrome tables

The error correction coding procedure is as follows:

1. Use $S = rH^T$ to calculate the syndrome of r
2. Find out common first item (error pattern) e_j , its syndrome equals rH^T
3. This error pattern is supposed to be the error caused by channel
4. The identified receive correction vector or code word equals $U = r + e_j$

III. BLOCK PRODUCT TURBO CODE DESIGN PRINCIPLE

The 96 messages must be arranged in 11×9 matrix form before BPTC coding, the 3 absent messages are set as 0, placed in the last three positions of the matrix (as shown in Fig. 5),

Step 3. Hamming(15,11) encoder											Step 1. Message [96]				
H_R0(3)	H_R0(2)	H_R0(1)	H_R0(0)	I(10)	I(9)	I(8)	I(7)	I(6)	I(5)	I(4)	I(3)	I(2)	I(1)	I(0)	
H_R1(3)	H_R1(2)	H_R1(1)	H_R1(0)	I(21)	I(20)	I(19)	I(18)	I(17)	I(16)	I(15)	I(14)	I(13)	I(12)	I(11)	
H_R2(3)	H_R2(2)	H_R2(1)	H_R2(0)	I(32)	I(31)	I(30)	I(29)	I(28)	I(27)	I(26)	I(25)	I(24)	I(23)	I(22)	
H_R3(3)	H_R3(2)	H_R3(1)	H_R3(0)	I(43)	I(42)	I(41)	I(40)	I(39)	I(38)	I(37)	I(36)	I(35)	I(34)	I(33)	
H_R4(3)	H_R4(2)	H_R4(1)	H_R4(0)	I(54)	I(53)	I(52)	I(51)	I(50)	I(49)	I(48)	I(47)	I(46)	I(45)	I(44)	
H_R5(3)	H_R5(2)	H_R5(1)	H_R5(0)	I(65)	I(64)	I(63)	I(62)	I(61)	I(60)	I(59)	I(58)	I(57)	I(56)	I(55)	
H_R6(3)	H_R6(2)	H_R6(1)	H_R6(0)	I(76)	I(75)	I(74)	I(73)	I(72)	I(71)	I(70)	I(69)	I(68)	I(67)	I(66)	
H_R7(3)	H_R7(2)	H_R7(1)	H_R7(0)	I(87)	I(86)	I(85)	I(84)	I(83)	I(82)	I(81)	I(80)	I(79)	I(78)	I(77)	
H_R8(3)	H_R8(2)	H_R8(1)	H_R8(0)	R(2)	R(1)	R(0)	I(95)	I(94)	I(93)	I(92)	I(91)	I(90)	I(89)	I(88)	
H_C10(9)	H_C13(0)	H_C12(0)	H_C11(0)	H_C10(0)	H_C9(0)	H_C8(0)	H_C7(0)	H_C6(0)	H_C5(0)	H_C4(0)	H_C3(0)	H_C2(0)	H_C1(0)	H_C0(0)	
H_C14(1)	H_C13(1)	H_C12(1)	H_C11(1)	H_C10(1)	H_C9(1)	H_C8(1)	H_C7(1)	H_C6(1)	H_C5(1)	H_C4(1)	H_C3(1)	H_C2(1)	H_C1(1)	H_C0(1)	
H_C14(2)	H_C13(2)	H_C12(2)	H_C11(2)	H_C10(2)	H_C9(2)	H_C8(2)	H_C7(2)	H_C6(2)	H_C5(2)	H_C4(2)	H_C3(2)	H_C2(2)	H_C1(2)	H_C0(2)	
H_C14(3)	H_C13(3)	H_C12(3)	H_C11(3)	H_C10(3)	H_C9(3)	H_C8(3)	H_C7(3)	H_C6(3)	H_C5(3)	H_C4(3)	H_C3(3)	H_C2(3)	H_C1(3)	H_C0(3)	

Step 2. Zero padding [3]

Step 4. Hamming(13,9) encoder

Figure 5. BPTC (196, 96) Matrix

which are R(0), R(1) and R(2), one R(3) is added in the coding process for replacing the position during interleaving after coding, so it is not placed in the messages to be encoded.

A. BPSK BPTC Communication System Architecture

The digital signal will be coded by BPTC (196, 96) and modulated by binary phase shift key (BPSK) modulator before transmission. When the signal passed through the AWGN channel, the received signal is demodulated by BPSK demodulator and sent to the decoder of BPTC (196, 96). The system block diagram is shown in Fig. 6.

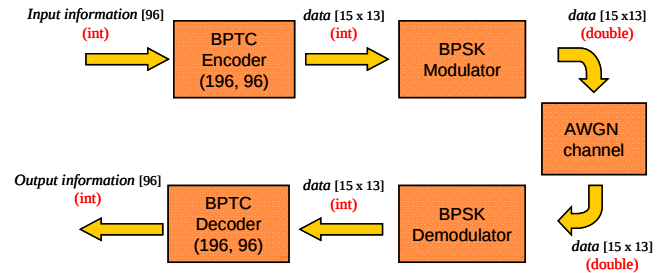


Figure 6. BPSK BPTC system block diagram

B. BPTC Encoding

The Hamming (15, 11) is encoded according to the above Hamming encoder, I (0) to I (10) are a set of messages, H_R0 (0), H_R0 (1), H_R0 (2), H_R0 (3), I (11) to I (21) derived from encoding are a set of messages, H_R1 (0), H_R1 (1), H_R1 (2) and H_R1 (3) are derived from encoding, till I (88) to R (2) are a set of messages, afterwards, Hamming (15, 11) encoding is finished.

After Hamming (15, 11) was encoded, the message array has changed to 15×9 form, and then Hamming (13, 9) will be encoded.

Encode Hamming (13, 9), I (0), I (11), I (22),..., I (88) are a set of messages, H_C0 (0), H_C0 (1), H_C0 (2), H_C0 (3) are derived from encoding, I (1), I (12), I (23),..., I (89) are a set of messages, H_C1 (0), H_C1 (1), H_C1 (2) and H_C1 (3) are derived from encoding, the rest can be deduced accordingly, till H_R0 (3), H_R1 (3), H_R2 (3),..., H_R8 (3) are a set of messages, afterwards, Hamming (13, 9) encoding is finished.

When the row and column encoding is completed, the matrix has changed to 15×13 form, and then the interleaving is carried out, due to the interleaving needs the sequence number, the encoded matrix must be renumbered, H_R0 (0) should be the 12th message, H_R0 (1) is the 13th message, and I (11) will be the 16th message, the rest can be deduced accordingly, the results are shown in Fig. 7.

Bit	Index	Interleave Index	Bit	Index	Interleave Index	Bit	Index	Interleave Index
I(0)	0	0	I(50)	66	74	H_R8(1)	132	148
I(1)	1	13	I(51)	67	87	H_R8(2)	133	161
I(2)	2	26	I(52)	68	100	H_R8(3)	134	174
I(3)	3	39	I(53)	69	113	H_C0(0)	135	187
I(4)	4	52	I(54)	70	126	H_C1(0)	136	4
I(5)	5	65	H_R4(0)	71	139	H_C2(0)	137	17
I(6)	6	78	H_R4(1)	72	152	H_C3(0)	138	30
I(7)	7	91	H_R4(2)	73	165	H_C4(0)	139	43
I(8)	8	104	H_R4(3)	74	178	H_C5(0)	140	56
I(9)	9	117	I(55)	75	191	H_C6(0)	141	69
I(10)	10	130	I(56)	76	8	H_C7(0)	142	82
H_R0(0)	11	143	I(57)	77	21	H_C8(0)	143	95
H_R0(1)	12	156	I(58)	78	34	H_C9(0)	144	108
H_R0(2)	13	169	I(59)	79	47	H_C10(0)	145	121
H_R0(3)	14	182	I(60)	80	60	H_C11(0)	146	134
I(11)	15	195	I(61)	81	73	H_C12(0)	147	147
I(12)	16	12	I(62)	82	86	H_C13(0)	148	160
I(13)	17	25	I(63)	83	99	H_C14(0)	149	173
I(14)	18	38	I(64)	84	112	H_C0(1)	150	186
I(15)	19	51	I(65)	85	125	H_C1(1)	151	1
I(16)	20	64	H_R5(0)	86	138	H_C2(1)	152	16
I(17)	21	77	H_R5(1)	87	151	H_C3(1)	153	29
I(18)	22	90	H_R5(2)	88	164	H_C4(1)	154	42
I(19)	23	103	H_R5(3)	89	177	H_C5(1)	155	55
I(20)	24	116	I(66)	90	190	H_C6(1)	156	68
I(21)	25	129	I(67)	91	7	H_C7(1)	157	81
H_R1(0)	26	142	I(68)	92	20	H_C8(1)	158	94
H_R1(1)	27	155	I(69)	93	33	H_C9(1)	159	107
H_R1(2)	28	168	I(70)	94	46	H_C10(1)	160	120
H_R1(3)	29	181	I(71)	95	59	H_C11(1)	161	133
I(22)	30	194	I(72)	96	72	H_C12(1)	162	146
I(23)	31	11	I(73)	97	85	H_C13(1)	163	159
I(24)	32	24	I(74)	98	98	H_C14(1)	164	172
I(25)	33	37	I(75)	99	111	H_C0(2)	165	185
I(26)	34	50	I(76)	100	124	H_C1(2)	166	2
I(27)	35	63	H_R6(0)	101	137	H_C2(2)	167	15
I(28)	36	76	H_R6(1)	102	150	H_C3(2)	168	28
I(29)	37	89	H_R6(2)	103	163	H_C4(2)	169	41
I(30)	38	102	H_R6(3)	104	176	H_C5(2)	170	54
I(31)	39	115	I(77)	105	189	H_C6(2)	171	67
I(32)	40	128	I(78)	106	6	H_C7(2)	172	80
H_R2(0)	41	141	I(79)	107	19	H_C8(2)	173	93
H_R2(1)	42	154	I(80)	108	32	H_C9(2)	174	106
H_R2(2)	43	167	I(81)	109	45	H_C10(2)	175	119
H_R2(3)	44	180	I(82)	110	58	H_C11(2)	176	132
I(33)	45	193	I(83)	111	71	H_C12(2)	177	145
I(34)	46	10	I(84)	112	84	H_C13(2)	178	158
I(35)	47	23	I(85)	113	97	H_C14(2)	179	171
I(36)	48	36	I(86)	114	110	H_C0(3)	180	184
I(37)	49	49	I(87)	115	123	H_C1(3)	181	1
I(38)	50	62	H_R7(0)	116	136	H_C2(3)	182	14
I(39)	51	75	H_R7(1)	117	149	H_C3(3)	183	27
I(40)	52	88	H_R7(2)	118	162	H_C4(3)	184	40
I(41)	53	101	H_R7(3)	119	175	H_C5(3)	185	53
I(42)	54	114	I(88)	120	188	H_C6(3)	186	66
I(43)	55	127	I(89)	121	5	H_C7(3)	187	79
H_R3(0)	56	140	I(90)	122	18	H_C8(3)	188	92
H_R3(1)	57	153	I(91)	123	31	H_C9(3)	189	105
H_R3(2)	58	166	I(92)	124	44	H_C10(3)	190	118
H_R3(3)	59	179	I(93)	125	57	H_C11(3)	191	131
I(44)	60	192	I(94)	126	70	H_C12(3)	192	144
I(45)	61	9	I(95)	127	83	H_C13(3)	193	157
I(46)	62	22	R(0)	128	96	H_C14(3)	194	170
I(47)	63	35	R(1)	129	109	R(3)	195	183
I(48)	64	48	R(2)	130	122			
I(49)	65	61	H_R8(0)	131	135			

Figure 7. Interleaving indices for BPTC (196, 96)

The index in Fig. 7 is the redefined message number, multiply the message number by 13, divided by 196, take the remainder, the new interleave index number can be formulated as the following statement

$$\text{Interleave Index} = \text{Index} \times 13 \text{ modulo } 196$$

The result of interleaving can be obtained from the new numbers derived from the above equation; the messages should be transmitted according to the new number sequence.

It should be paid special attention to that the new number of R (3) which was not placed in the messages originally is placed in the position of 183 after interleaving, and the I (11) in the original messages is placed in the position of 195 after interleaving, only the messages of new numbers from 0 to 194 will be sent out, the position of 195, i.e. the original I (11) will not be transmitted together.

C. BPTC Decoding

The messages received by the receiving terminal are in the form of 15×13 matrix, de-interleaving must be carried out before decoding to restore the original message sequence. Decoding can be carried out after de-interleaving, the decoding should be in such an order that encoded late should be decoded first, that encoded early should be decoded late.

Decode Hamming (13, 9) according to the aforesaid Hamming decode, I (0), I (11), I (22),..., H_C0 (2) and H_C0 (3) after sequence restoration are a set of messages, decode and correct I (0), I (11), I (22),..., I (88), I (1), I (12), I (23),..., H_C1 (2) and H_C1 (3) are a set of messages, decode and correct I (1), I (12), I (23),..., I (89), the rest can be deduced accordingly, till H_R0 (3), H_R1 (3), H_R2 (3),..., H_C14 (2) and H_C14 (3) are a set of messages, afterwards, Hamming (13, 9) decoding is finished.

After Hamming (13, 9) decoding, the message array has changed to 15×9 form, and then Hamming (15, 11) will be decoded.

Decode Hamming (15, 11), I (0), I (1), I (2),..., H_R0 (2) and H_R0 (3) are a set of messages, decode and correct I (0), I (1), ..., I (10). I (11), I (12), I (13),..., H_R1 (2) and H_R1 (3) are a set of messages, decode and correct I (11), I (12), ..., I (21), the rest can be deduced accordingly, till I (88) to H_R8 (3) are a set of messages, afterwards, Hamming (15, 11) decoding is finished.

D. BPTC error correcting mechanism analysis

Although the BPTC is composed of quadratic Hamming code, its probability of being corrected in the second dimension coding is increased by using the fundamental characteristics of turbo code.

As shown in Fig. 8 (a), if the column vectors only have error in position (1.), they can be certainly corrected by using the characteristics of Hamming. When the column vectors have error in position (2.), if the row vectors of the second

dimension have no error, the error in position (2.) can be corrected as well, the key is whether the second dimension has error in position (3.), the same situation can be observed in any column, therefore, return to the column vectors, although the first dimension can correct only one message, the other messages can be corrected in the second dimension probably.

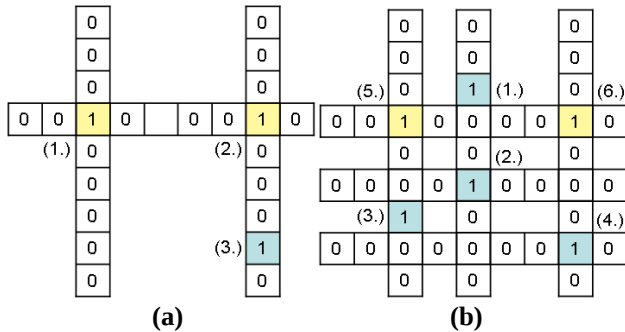


Figure 8. Schematic diagram of two-dimension correction capability

For example, in Fig. 8 (b), when decoding the column vectors of the first dimension, positions (1.), (2.), (3.), and (4.) have only one error message, so these errors will certainly be corrected to the original messages. When decoding the row vectors of the second dimension, due to positions (5.) and (6.) can be probably corrected in the first dimension decoding, there will be three situations at this point, the first one is when the position (5.) is corrected in the first dimension decoding, the position (6.) will be corrected in the second dimension decoding. The second situation is when the position (6.) is corrected in the first dimension; the position (5.) will be corrected in the second dimension. The third situation is when the positions (5.) and (6.) cannot be corrected in the first dimension, the row vector decoding of the second dimension will still correct the errors in positions (5.) and (6.), for row vectors, (5.) and (6.) are one single error message, they can be corrected certainty.

E. BPSK BPTC C program optimization efficiency analysis

	†	-	*	%	† (double)	- (double)	* (double)	/ (double)	>> <<	&	^	log	sqrt	compare
BPTC_encoder	1182	1182	1380	195					7092	6858	384			6144
BPTC_decoder	1030	1022	1360	195					2915	2402	144			1944
BPSK_modulator														1
BPSK_demodulator														1
AWGN					2	2	7	4				1	2	1

Figure 9. BPSK BPTC optimization efficiency statistics

After calculation, the fundamental operations and functional equations are shown in Fig. 9., the double represents floating-point number operation, >> and << represent bit right shift and left shift respectively, compare instead of judgment operation and the compare represents identification operation. Most of quadratic square operations are changed into right shift and left shift, due to one left shift of number 1 in C code is equivalent to first power of 2, but the efficiency of left shift and right shift is much better than that of power, such conversion can improve the computing speed.

IV. GOLAY (20,8) CODE DESIGN PRINCIPLE

This paper also discusses Golay (20, 8) which is one type of block code. The minimum Hamming distance of general Golay (23, 12) code [6] [7] $d_{min} = 7$, number of corrected error bits $t = 3$. All the 3-bit errors can be corrected, so that Golay (23, 12) code is the so-called perfect code [8], for it meets the Hamming bound of the following equation

$$2^{n-k} = \sum_{i=0}^t \binom{n}{i}$$

The extended non-perfect Golay (24,12) code emphasizes the relation among n , k and t , namely, the coding rate can have simple integer ratio for convenient digital circuit implementation.

For the Golay (20, 8) code discussed in this paper, code word length after coding $n = 20$, the number of information bits $k = 8$, it is of extended Golay code. The minimum Hamming distance of this Golay (20, 8) code is increased to $d_{min} = 8$, besides correcting all 3-bit errors, the error correcting capability is improved to partial 4-bit errors. However, even the larger minimum Hamming distance $d_{min} = 8$ is not emphasized, the correction capability is improved by increasing the minimum Hamming distance, so as to reduce the signal error probability.

The predominance of Golay (20, 8) code is the large corresponding syndrome list, therefore, more error patterns can be corrected, certainly its decoding circuit is complicated relatively. The operating principle and simulation result of Golay (20, 8) code will be detailed below.

In the following equation, $g(x)$ is the generator polynomial of Golay (20, 8) code.

$$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1 = 6165_8$$

The generator matrix in Fig. 10 can be divided into two parts, the left part is an 8×8 identity matrix, and the right part is an 8×12 parity matrix.

1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	0	1	0
0	1	0	0	0	0	0	0	0	1	1	0	1	1	0	0	1	1	0	0	1
0	0	1	0	0	0	0	0	0	0	1	1	0	1	1	0	0	1	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	1	0	0	1	1	1
0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0
0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	1	0
0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	0	0	1	0	1	1
0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	1	1	1

Figure 10. The generator matrix of Golay (20, 8)

A. Golay (20, 8) encoding

The information of the original data of Golay (20, 8) is 8-bit, while added with 12-bit parity bit for coding; it changes into 20-bit code word transmission after coding. The transmitting terminal uses generator matrix for coding, the 1×8 information data matrix is multiplied by 8×20 generator matrix in Table 1 to obtain 1×20 coding code word matrix.

B. Golay (20, 8) decoding

The receiving terminal uses parity check matrix for decoding, the structure can be set as

$$H = [I_{n-k} \mid P^T]$$

Thus, the columns of generator matrix (G) are orthogonal to the columns of parity check matrix (H), i.e.

$$GH^T = 0$$

Therefore, HT can be expressed as

$$H^T = \begin{bmatrix} I_{n-k} \\ P \end{bmatrix}$$

Next, the decoding is divided into four steps:

Step1. Multiply received code word (1×20) by H^T to obtain (1×12) syndrome vector,

Step2. The only error pattern was deduced out by the corresponded syndrome. The vector matrix of error pattern is (1×20),

Step3. Add this error pattern to the received codeword, so as to obtain the corrected code word,

Step4. Since the Golay (20, 8) code is of systematic code, we can take out the original information from the corrected code word, see the following equation,

$$[\text{received codeword}]_{1 \times 20} = [M_1 \ M_2 \ \dots M_8 \ P_1 \ P_2 \ P_3 \ \dots P_{12}]$$

1×20

therefore, the decoded information is $[M_1 \ M_2 \ M_3 \ \dots M_8]$.

C. MSK Golay communication system architecture

The digital signal will be coded by Golay (20, 8) and modulated by minimum shift key (MSK) modulator before transmission. When the signal passed through the AWGN channel, the received signal is demodulated by MSK demodulator and sent to the decoder of Golay (20, 8). The system block diagram is shown in Fig. 11.

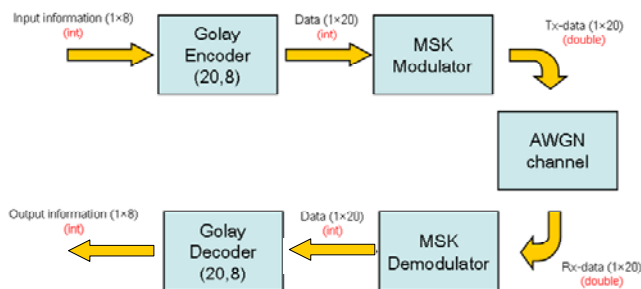


Figure 11. MSK Golay system block diagram

First, in C-code program the transmission signal of a block of 1×8 bits is changed into an int, enters the Golay encoder for

coding (here also announced as int type), it is a 1×20 block codeword after coding, and then enters MSK modulator (here the data type is int input, double output), AWGN channel (data type is equivalent to MSK modulator), MSK demodulator (data type is equivalent to MSK modulator), the received signal of a block information bits of 1×8 bits is changed into an int output after decoding.

The bit error rate (BER) is figured out by comparing the original transmission signal with the finally decoded received signal.

The program must be optimized in order to reduce operation steps and simplify the functional equation. The Box-Muller [9] is used for the generation principle of AWGN noise, the equation is as follows

$$\sqrt{-2\ln(U_1)} \cos(2\pi U_2)$$

Among which, U_1 and U_2 are random number of 0 to 1, we should use the program produced cos function look-up table rather than the C language built-in function, so as to increase the computing speed for program optimization.

This paper uses a simpler equation to replace cos function for program optimization. The equation is as follows.

$$2.0 * \text{rand}() / \text{RAND_MAX} - 1.0$$

Among which, $\text{rand}() / \text{RAND_MAX}$ is a random number of 0 to 1, $\text{RAND_MAX} = 32767$, image this range to the range of cos function, changes the original range of 0 to 1 to -1 to 1.

Fig. 12 is AWGN in Rayleigh probability density function (PDF) profile, the blue solid curve is theoretical value, the red dotted curve is the simulation result of C language optimized program, here, the SNR is 20dB.

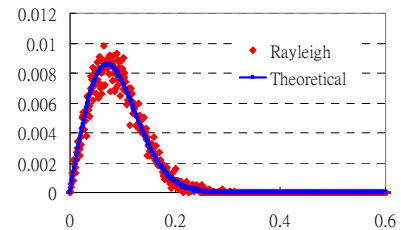


Figure 12. The AWGN in Rayleigh probability density function profile

D. Golay (20,8) encoding processes

The following (1×8) matrix is the data information.

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Multiply a set of data information of matrix (1×8) by generator matrix in Fig. 9 to obtain the coded code word (1×20)

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

E. Golay (20,8) decoding processes

The following 20×12 transposed parity check matrix (H^T) is used for decoding.

0	0	1	1	1	1	0	1	1	0	1	0
1	1	0	1	1	0	0	1	1	0	0	1
0	1	1	0	1	1	0	0	1	1	0	1
0	0	1	1	0	1	1	0	0	1	1	1
1	1	0	1	1	1	0	0	0	1	1	0
1	0	1	0	1	0	0	1	0	1	1	1
1	0	0	1	0	0	1	1	1	1	1	0
1	0	0	0	1	1	1	0	1	0	1	1
1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	1

Step1. Multiply the coded code word by transposed parity check matrix to obtain syndrome (1×12)

0	1	0	0	1	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---

Step2. List the syndromes corresponding to error patterns

From the look-up table of $t = 1, 2, 3$ [10], each of the 1350 error patterns corresponds to a unique syndrome, in other words, 1-bit to 3-bit errors can be corrected completely. The original error pattern is that the number of combinations of four error bits in twenty code word bits is 4845. However, as 12 bits are a group in syndrome, there are $2^{12} = 4096$ syndromes. Therefore, when the information has 4 bit errors, only part of them can be corrected.

Step3. Use the syndrome list to find out the corresponding error pattern, add the error pattern to code word, so as to correct 1-bit to 3-bit errors and partial 4-bit errors and reduce the error probability.

F. MSK Golay C program optimization efficiency analysis

	+	-	*	+(double)	-(double)	*(double)	/(double)	>>	<<	&	^	log	sqrt	cos	compare
Golay_encoder	20	20	20					420	420	420					400
Golay_decoder		12	12					253	241	253					240
MSK_modulator				400		1600								800	1
MSK_demodulator				1200		2400								800	1
AWGN				2	2	7	4					1	2		1

Figure 13. MSK Golay optimization efficiency statistics

Fig. 13 shows the fundamental operations and functional equations used in the system program after optimization, the double represents the floating-point number operation, >> and << represent bit right shift and left shift respectively, log and sqrt represent log and square root in mathematical operation

respectively, the compare represents judgment operation of if, switch or ?.

According to this table, power operation is avoided when composing the program, the power of 2 is replaced by left shift, for example $1 \ll 1$ is 2 to the first power, $1 \ll 2$ is 2 to the second power, thus, the program can be optimized and realized on DSP easily.

V. COMPARISON OF C CODE ALGORITHM SIMULATION

When we simulate a communication system based on BPSK modulation and channel AWGN [11], and keep the error probability at 10^{-5} , the power consumed is 9.5 dB (as shown in Fig. 14). Hamming (15, 11) and BPTC (196, 96) are compared in the same AWGN channel environment, disregarding the excess bandwidth consumed by coding, if the error probability should be kept at 10^{-5} as well, the BPTC has about 0.6 dB coding gain compared with Hamming code.

If the E_b/N_0 must be kept at 8 dB in the same AWGN channel environment, the error probability of BPTC is 0.000006140625, and the error probability of Hamming (15, 11) is 0.0000299, it means the error probability of BPTC is lower than that of Hamming code by about 10^{-1} at the same transmission power.

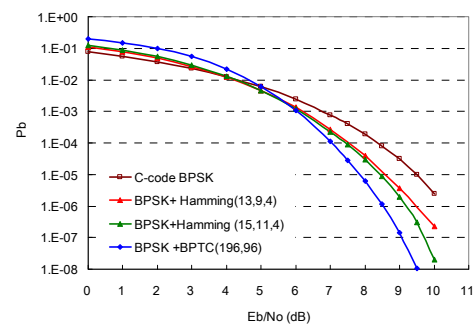


Figure 14. BPTC and Hamming error probability curves

Fig 15 shows the MSK error probability curves calculated by theoretical MSK (blue solid) and simulated by C code (Red dotted). When $P_b = 10^{-5}$, the theoretical MSK E_b/N_0 is at 12.6dB, and the C code MSK E_b/N_0 is at 12.5dB, obviously, the two curves are approximately identical.

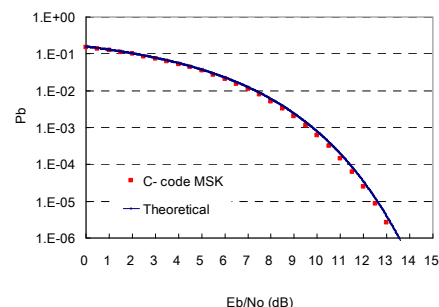


Figure 15. MSK error probability curves simulated by C code (Red dotted)

Fig 16 shows the error probability curves without coding and with Golay (20, 8) coding, when $P_b = 10^{-5}$, the uncoded E_b/N_0 is about 12.5dB, the Golay (20, 8) of $t = 4$ is about 11.2dB. It is observed that at the same error probability, the coded E_b/N_0 is decreased by 1.3dB, the coding gain is 1.3dB, so the advantage of taking Golay (20, 8) is quite obvious.

Afterwards, the Golay (20, 8) of $t = 4$ in Figure 15 is compared with the Golay (20, 8) curve with correction capability up to $t = 3$ only. At $P_b = 10^{-5}$, Golay (20, 8) of $t = 4$ is about 11.2dB, and the Golay (20, 8) of $t = 3$ is 12.2dB, the difference is 1dB. Therefore, the Golay (20, 8) of $t = 3$ has not reached the due code efficiency. Because there should be 4096 syndromes for correcting 4096 error patterns correspondingly, actually, there are only 1350 syndromes when $t = 3$, the error patterns that can be corrected are decreased by 67.04%.

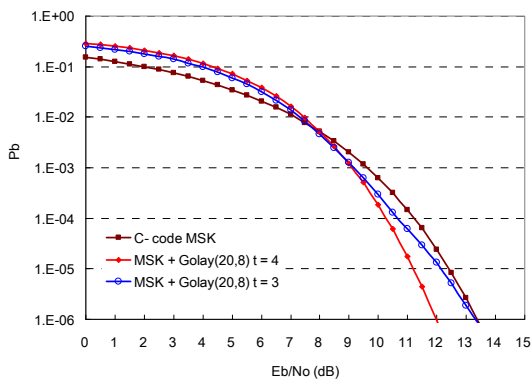


Figure 16. The error probability curves without and with Golay (20, 8) code

In addition, the E_b/N_0 thresholds of BPSK Golay (20, 8) of $t = 4$ (green and orange) and MSK Golay (20, 8) of $t = 4$ (pink and brown) are compared in Fig 17. The E_b/N_0 threshold of BPSK Golay (20, 8) is at 5.2dB; nevertheless, MSK Golay (20, 8) is at 8dB. There is a 2.8dB difference between them. That indicates BPSK Golay (20, 8) is more robust than MSK Golay (20, 8) for 2.8dB against noise. At same bit error probability $P_b = 10^{-5}$, MSK Golay (20,8) of $t = 4$ E_b/N_0 is 11.2dB, yet, BPSK Golay (20, 8) of $t = 4$ E_b/N_0 is 8.4dB. There is a same 2.8dB difference between them.

With regard to Hamming (16, 11, 4) code, there is 1.2 dB lower E_b/N_0 threshold compared to Golay (20, 8) of $t = 4$ in both BPSK (4dB) and MSK (6.8dB). According to this result, we infer that Hamming (16, 11, 4) is more solid than Golay (20, 8) of $t = 4$ against noise and getting better performance in larger E_b/N_0 .

At last, we compare the E_b/N_0 thresholds of BPSK BPTC (196, 96) (red and pink) and MSK Golay (20, 8) of $t = 4$ (dark blue and brown) in Fig 18. The E_b/N_0 threshold of BPSK BPTC (196, 96) is at 5dB; nevertheless, MSK Golay (20, 8) of $t = 4$ is at 8dB. There is a 3dB difference between them. That indicates BPSK BPTC (196, 96) is 3dB more robust than MSK Golay (20, 8) of $t = 4$ and only 0.2dB more robust than BPSK Golay (20, 8) of $t = 4$. From Fig. 18, we can also observe that BPSK BPTC (196, 96) is getting better performance when the signal E_b/N_0 exceeding the cross over point and approaching 0.5dB gain relative to BPSK Golay (20, 8) of $t = 4$.

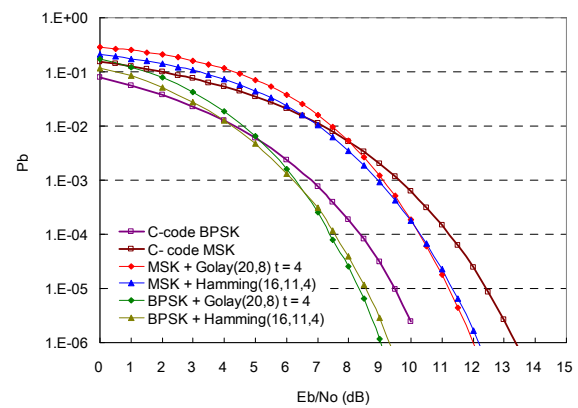


Figure 17. Comparison of the E_b/N_0 threshold of BPSK Golay (20, 8) of $t = 4$ (green and orange) and MSK Golay (20, 8) of $t = 4$ (pink and brown)

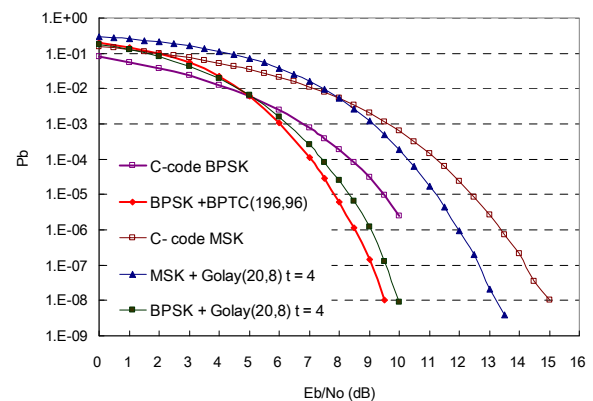


Figure 18. Comparison of the E_b/N_0 threshold of BPSK BPTC (196, 96) (red and pink) and MSK Golay (20, 8) of $t = 4$ (dark blue and brown)

VI. CONCLUSIONS AND FUTURE PROSPECTS

This paper observes the efficiency of BPTC, and compares the complexity and coding gain with Hamming code, and the physical mechanism of decoding efficiency difference between Hamming code and BPTC.

We can infer in this study that BPSK BPTC (196, 96) is a better choice than BPSK Golay (20, 8) of $t = 4$, if signal E_b/N_0 is stronger than 5dB; nevertheless, BPSK Golay (20, 8) of $t = 4$ will be a better choice than BPSK Hamming (16, 11, 4).

In order to improve the communication quality at limited transmission power, it is observed from Figure 16 in this paper that when E_b/N_0 is at 8dB, the uncoded MSK P_b is 0.00519; MSK Golay (20, 8) is 0.00526. Once the transmission power reaches the certain threshold, the Golay (20, 8) $t = 4$ coded error probability curves have obvious difference with the uncoded one. For example, when a communication quality is required $P_b = 10^{-5}$, the uncoded MSK E_b/N_0 is 12.5dB, the MSK Golay (20, 8) $t = 4$ is 11.2dB. Therefore, the coded E_b/N_0 is decreased by 1.3dB and the coding gain is 1.3dB in the same communication quality.

Although, the E_b/N_0 threshold of BPSK Golay (20, 8) is lower than the E_b/N_0 threshold of MSK Golay (20, 8), from Figure 17, we can see significantly that both coding benefits of

MSK Golay (20, 8) of $t = 4$ and BPSK Golay (20, 8) of $t = 4$ can only be achieved when the E_b/N_0 threshold of received signal are reached.

MSK modulation scheme has a great advantage of power efficiency in peak to average power ratio (PAPR), but less efficient in error performance compared to BPSK modulation scheme. At same bit error probability $P_b = 10^{-5}$, uncoded BPSK E_b/N_0 is about 9.6dB, there is a approximate 2.9dB gain compared to uncoded MSK. At the same power efficiency, $E_b/N_0 = 8.4$ dB, BPSK Golay (20, 8) of $t = 4$ can reach $P_b = 10^{-5}$, but, MSK can reach worse performance than $P_b = 10^{-3}$.

Besides the aforesaid discussions, this paper uses 4096 syndromes of Golay (20, 8) when $d_{min} = 8$, compares correcting partial 4-bit errors and correcting only 3-bit errors, it is observed that the error patterns that can be corrected are increased by 67.04% when 2^{12} syndromes are used completely.

This paper uses C language for simulation analysis, when the C code system program is constructed, the system program is optimized and realized on the DSP platform, fast verification and SDR concept can be realized by replacing modules, such as modulation modules, channel modules and coding modules, so that different modulation and encoding modes can adjust parameters in software directly according to different system specification requirements, so as to economize hardware cost and product development time.

The further goal of this implementation is to be applied to the wireless communication system of SDR concept, so that different modulation and encoding modes can adjust

parameters in firmware directly according to different system specification requirements.

REFERENCES

- [1] Brenard Sklar, "Digital Communications Fundamentals and Applications", Second edition, 2001.
- [2] Shu Lin, Daniel J. Costello, Jr, "Error Control Coding", Second edition, 1983.
- [3] errou, C., Glavieux, A., and Thitimajshima, P. "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," IEEE Pcoceedings of the Int. Conf. on Commuications, Geneva, Switzerland, May 1993 (ICC '93), pp. 1064-1070.
- [4] Berrou, C. and Glavieux, A. "Near Optimum Error-Correcting Coding and Decoding: Turbo Codes," IEEE Trans. On Communications, vol. 44, no. 10, October 1996, pp.1261-1271.
- [5] Odenwalder, J. P., Error Control Coding Handbook, Linkabit Corporation, San Diego, Calif., July 15, 1976.
- [6] M. J. E. Golay, "Notes on Digital Coding," Proc. IRE, vol. 37, no. 6, pp. 657, 1949.
- [7] F. J. MacWilliams and N. J. A. Sloane, "The Theory of Error-Correcting Codes", North Holland, Amsterdam, 1977.
- [8] Zong Cheng, Chang, "Principles and Applications of Error-correcting Codes", Publishing House of Electronics Industry, China, 2003.
- [9] Takashi Shinzato, "Box Muller Method" January 27, 2007.
- [10] Yi Hua Chen, Jue Hsuan Hsiao, Pang-Fu Liu, Jheng-Shyuan He, "Golay (20, 8) C Code Simulation and Implementation in DSP Chip", Proceedings of International Conference on Consumer Electronics, Communications and Networks. Volume 6, pp. 4990-4994, 2011.
- [11] Yi Hua Chen, Jue Hsuan Hsiao, Kun Feng Lin, Chun Chun Yeh, "BPTC(196,96) Code Simulation and Implementation in C-code DSP chip", Proceedings of International Conference on Consumer Electronics, Communications and Networks. Volume 6, pp. 4833-4837, 2011.